Towards Generating Query to Perform Query Focused Abstractive Summarization using Pre-trained Model

Deen Mohammad Abdullah and **Yllias Chali**
University of Lethbridge
Lethbridge, AB, Canada
{deen.abdullah, yllias.chali}@uleth.ca

Abstract

Query Focused Abstractive Summarization (*QFAS*) represents an abstractive summary from the source document based on a given query. To measure the performance of abstractive summarization tasks, different datasets have been broadly used. However, for *QFAS* tasks, only a limited number of datasets have been used, which are comparatively small and provide single sentence summaries. This paper presents a query generation approach, where we considered most similar words between documents and summaries for generating queries. By implementing our query generation approach, we prepared two relatively large datasets, namely CNN/DailyMail and Newsroom which contain multiple sentence summaries and can be used for future *QFAS* tasks. We also implemented a pre-processing approach to perform *QFAS* tasks using a pre-trained language model, *BERTSUM*. In our pre-processing approach, we sorted the sentences of the documents from the most query-related sentences to the less query-related sentences. Then, we fine-tuned the *BERTSUM* model for generating the abstractive summaries. We also experimented on one of the largely used datasets, Debatepedia, to compare our *QFAS* approach with other models. The experimental results show that our approach outperforms the state-of-the-art models on three ROUGE scores.

1 Introduction

Text summarization has two major types: extractive summarization and abstractive summarization. The extractive summarization approach only selects important sentences for generating extractive summaries and may lose the main context of the documents. In contrast, the abstractive summarization approach considers all the sentences of the document to hold the actual context of the document and paraphrase sentences to generate abstractive summaries. Query focused abstractive summarization (*QFAS*) emphasizes those sentences relevant to the given query and generates abstractive summaries based on the query. For example, a user may need to know the summary of the tourist places located in Vancouver rather than all the tourist places of the entire Canada. Then the *QFAS* approach will focus on the query keywords 'tourist places' 'Vancouver'and generate an abstractive summary.

With the advancement of the neural network, modern approaches of text summarization have focused on abstractive summarization, which paraphrases the words in the sentences by using encoder-decoder architecture (Rush et al., 2015; Nallapati et al., 2016; See et al., 2017; Tan et al., 2017; Narayan et al., 2018). With the RNN encoder-decoder model, Hu et al. (2015) introduced a dataset for Chinese text summarization. To solve the problem of recurring words in encoder-decoder models, Chen et al. (2016) have given an attention model to minimize the repetition of the same words and phrases.

In other works, the transformer model has been used to get better summaries (Egonmwan and Chali, 2019). A pretrained language model, Bidirectional Encoder Representations from Transformers (*BERT*) (Devlin et al., 2019) model can combine the word and sentence representations in a single substantial transformer (Vaswani et al., 2017), which can be fine-tuned for next sentence prediction tasks. Recently, the *BERT* has been used on the *BERTSUM* model for summarization tasks and showed state-of-the-art results (Liu and Lapata, 2019). However, all these research works were focused only on generating better abstractive summaries and did not consider the relevance of query for abstractive summarization.

Query focused summarization highlights those sentences which are relevant to the context of a

Proceedings of The 13th International Conference on Natural Language Generation, pages 80–85,
Dublin, Ireland, 15-18 December, 2020. ©2020 Association for Computational Linguistics

given query. Still, only few works have been done on *QFAS* (Nema et al., 2017; Hasselqvist et al., 2017; Aryal and Chali, 2020). Here, Nema et al. (2017), Aryal and Chali (2020) independently used Debatepedia[1] (Nema et al., 2017) dataset and Hasselqvist et al. (2017) used CNN/DailyMail[2] (Hermann et al., 2015) dataset for *QFAS* tasks. Debatepedia dataset is a small dataset which consists of single sentence summaries. Hence, we intended to investigate whether relatively large datasets with multiple sentence summaries perform better on *QFAS* tasks. We prepared and used two large datasets; CNN/DailyMail and Newsroom[3] (Grusky et al., 2018) for our *QFAS* task, which have multiple sentence summaries. Using CNN/DailyMail dataset, Hasselqvist et al. (2017) generated queries for *QFAS* task and conducted their research experiments. In their query generation approach, the authors considered only summaries and did not focus on the relevant documents which may have an impact on the performance of their proposed model. Therefore, we developed our new query generation approach, considering the relevant documents and summaries for our *QFAS* task.

In our *QFAS* approach, we emphasized on the input representation and implemented our idea of sorting the sentences of the documents according to the corresponding queries. Then we used our pre-processed input to fine-tune the *BERTSUM* model for generating abstractive summaries. For CNN/DailyMail our approach achieved better ROUGE scores than the work of Hasselqvist et al. (2017). As there is no previous work which performs *QFAS* tasks on Newsroom dataset, we present our results for future research comparison. We also implemented our query generation and *QFAS* approaches on Debatepedia dataset and found that our approaches work well on Debatepedia dataset in comparison with the existing *QFAS* based research works.

2 Related Work

The research work of Nema et al. (2017) implemented the attention model for both queries and documents on Debatepedia dataset. Their model succeeded in solving the problem of repeating phrases in summaries. They proposed a model with two key addition to the encode attend decode model. In other work, Aryal and Chali (2020) focused on solving the problem of noisy encoder. The authors focused on representing the input sequence in a selective approach and used sequence-to-sequence model on Debatepedia dataset to generate query focused abstractive summaries. Hasselqvist et al. (2017) proposed a pointer-generator model for query focused abstractive summarization on CNN/DailyMail dataset. They incorporated attention and pointer generation mechanism on a sequence-to-sequence model.

To perform many natural language tasks pretrained language models have been used (Devlin et al., 2019). Introducing a novel document level encoder based on *BERT*, Liu and Lapata (2019) proposed a fine-tuning schedule and named the model as *BERTSUM* to generate summaries. For the decoding phase, they followed the same approach as Vaswani et al. (2017). But in their work, they did not consider the query relevance for the summarization. Therefore, we used the *BERTSUM* model as a pretrained language model for *QFAS* task. We pre-processed the input according to the query and then fine-tuned the *BERTSUM* model to generate query focused abstractive summaries.

3 Dataset Preparation

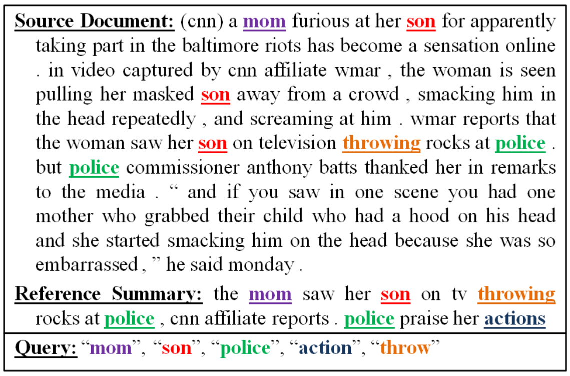
In this work, we used three datasets; CNN/DailyMail, Newsroom and Debatepedia for our experiments. For our *QFAS* task, we prepared these three datasets with our new query generation approach. The CNN/DailyMail dataset comprises of 287K news articles with 3-4 lines related highlights. In our work, we collected the stories as the text documents and the highlights as the corresponding summaries. In this way, the summaries contain more than one sentence which made the dataset more useful for the pretrained language models. The Newsroom dataset has been developed from 38 major news publications. The authors collected words and phrases from articles to generate summaries by combining the abstractive and extractive approaches. In Newsroom, there are three types of datasets: *Abstractive*, *Extractive*, and *Mixed*. For our *QFAS* task, we used *Abstractive* and *Mixed* datasets of Newsroom where we eliminated those data which had single sentence summaries. The Debatepedia dataset corpus has 663 debates under 53 categories. Though the dataset contains single sentence summaries, some *QFAS* models used this dataset

[1] https://github.com/PrekshaNema25/DiverstiyBasedAttentionMechanism

[2] https://cs.nyu.edu/~kcho/DMQA/

[3] http://lil.nlp.cornell.edu/newsroom/

for their experiments. Therefore, we experimented our query generation and *QFAS* approaches using the Debatepedia dataset to compare whether our new approach outperforms the state-of-the art result or not.



Source Document: (cnn) a **mom** furious at her **son** for apparently taking part in the baltimore riots has become a sensation online . in video captured by cnn affiliate wmar , the woman is seen pulling her masked **son** away from a crowd , smacking him in the head repeatedly , and screaming at him . wmar reports that the woman saw her **son** on television **throwing** rocks at **police** . but **police** commissioner anthony batts thanked her in remarks to the media . “ and if you saw in one scene you had one mother who grabbed their child who had a hood on his head and she started smacking him on the head because she was so embarrassed , ” he said monday .

Reference Summary: the **mom** saw her **son** on tv **throwing** rocks at **police** , cnn affiliate reports . **police** praise her **actions**

Query: “**mom**”, “**son**”, “**police**”, “**action**”, “**throw**”

Figure 1: Generated query from given document and summary in CNN/DailyMail

3.1 Our Query Generation Approach

When we search in a text with our given query, we expect the presence of those query keywords in our search results. In query focused summarization, both the generated summary as well as the source document should contain the context of the query keywords. For example, we have a document that contains a patient's medicine information corresponding to his/her different diseases. If the patient wants to know about his/her diabetes related medicine information as a summary and provide a query (‘diabetes’ ‘medicine ’), then the main document should contain the information on ‘diabetes’ and ‘medicine’. Otherwise, we can assume that the source document has no information regarding that person's diabetes related medicine, and both the document and the query will be considered as invalid. Similarly, in the summary, the presence of these two keywords will confirm that the generated summary is query relevant. The query holds the context of the summary, where the context of the query keywords should be present in the source document. For this reason, we considered those words from the summary that are most similar to the document.

In our query generation approach, we pre-processed each document and the document's corresponding summary. We performed tokenization, the removal of the stop words, and lemmatization as pre-processing steps. Then we used the Python library, *spaCy*[4] and trained a pretrained model ‘en_core_web_md’ with the source document. Then, we considered each word of the summary and calculated the cosine similarity with the trained model. Finally, we selected five most similar words as our query. In Figure 1, we have shown our generated query from a document and the corresponding summary for CNN/DailyMail. Here, we can observe that the word ‘action’ is not present in the source document but convey contextual relation with the document and hence selected as one of the query keywords.

4 Our Summarization Framework

We used the source document and query as the system input and generated summary as system output. Our summarization framework has two parts, at first we pre-processed the source document according to the query by which we incorporated the query relevance to our *QFAS* task. Then, we used the *BERTSUM* model to generate abstractive summaries, where we fine-tuned the model with our pre-processed source documents. Our summarization approach is shown in Figure 2.

4.1 Our Pre-Processing Approach

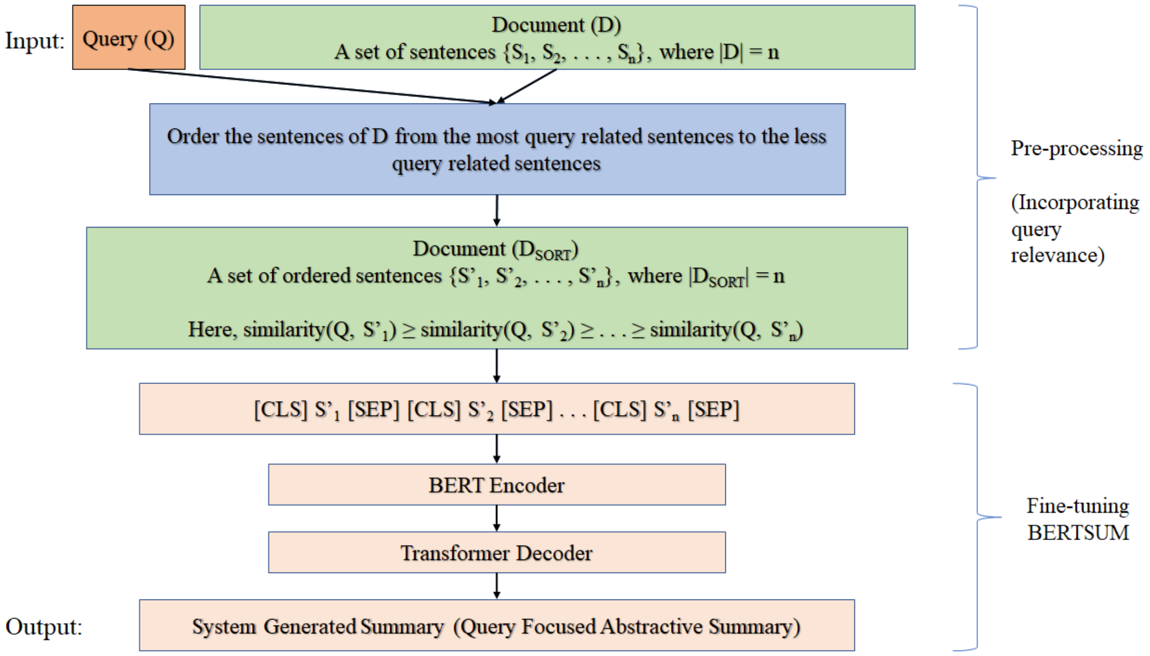
We sorted the sentences of a document according to the relevance of the generated query. Given a document, $D = \{S_1, S_2, ..., S_n\}$ and generated query, $Q = \{q_1, q_2, ..., q_m\}$, we ordered the sentences to get the sorted document, $D_{SORT} = \{..., S_i, S_j, ...\}$, where, $1 \leq i, j \leq n$; $i \neq j$; and $similarity(Q, S_i) \geq similarity(Q, S_j)$.

Here, we used the Python library, *spaCy* and trained ‘en_core_web_md’ model with the query. Then, for each sentence of the document, we calculated the cosine similarity with the trained model. Finally, we sorted the document from the most similar to the less similar values.

4.2 Fine-Tuning the BERTSUM Model

In this paper, we followed the same fine-tuning approach of Liu and Lapata (2019). We selected sorted sentences one by one from the document, D_{SORT} and tokenized each sentence by following the work of Durrett et al. (2016). Then, we incorporated the $[CLS]$ token at the beginning of each sentence and assigned three embeddings; token embedding, segmentation embedding, and position

[4] https://spacy.io/



Input:
Query (Q)
Document (D)
A set of sentences {S1, S2, . . . , Sn}, where |D| = n
Order the sentences of D from the most query related sentences to the less query related sentences
Document (DSORT)
A set of ordered sentences {S'1, S'2, . . . , S'n}, where |DSORT| = n
Here, similarity(Q, S'1) ≥ similarity(Q, S'2) ≥ . . . ≥ similarity(Q, S'n)
[CLS] S'1 [SEP] [CLS] S'2 [SEP] . . . [CLS] S'n [SEP]
BERT Encoder
Transformer Decoder
Output:
System Generated Summary (Query Focused Abstractive Summary)
Pre-processing
(Incorporating query relevance)
Fine-tuning BERTSUM

Figure 2: Our summarization approach (Pre-processing and Fine-tuning)

embedding for each token. Finally, the summation of three embeddings of the input document was passed to the transformer. Token embedding has been used to represent the meaning of each token, whereas segmentation embedding is used to identify each sentence separately. The position embedding has been used to determine the position of each token. Following the same encoder-decoder framework of See et al. (2017), we used pretrained encoder and 6-layered transformer for decoder as Liu and Lapata (2019) used for their *BERTSUM* model. We used Adam optimizers, $\beta_1 = 0.9$ for the encoder, and $\beta_2 = 0.999$ for the decoder to make our fine-tuning stable and used the learning rates for encoder and decoder as in following equations:

$$\alpha = \tilde{\alpha}.min(N^{-0.5}, N.warmup^{-1.5})$$

where, N stands for the iteration number, $warmup$ is $20,000$ for the encoder and $10,000$ for the decoder, and $\tilde{\alpha}$ is $2e^{-3}$ for the encoder and 0.1 for the decoder.

5 Experimental Setups

We implemented our query generation and *QFAS* approaches on CNN/DailyMail, Newsroom and Debatepedia datasets using the same experimental setup.

5.1 Implementation Details

We trained the model for $200,000$ steps on TITAN X GPU (GTX Machine) and used PyTorch (Paszke et al., 2017), OpenNMT (Klein et al., 2017). We imported 'bert-base-uncased' of the *BERT* (Devlin et al., 2019) model for utilizing the *BERTSUM* model. We set the dropout probability 0.1 and the label-smoothing factor 0.1 (Szegedy et al., 2016). For the encoder, we took 768 hidden units with the hidden size for feed-forward layers $2,048$. In the decoding phase, we used beam size 5, and tuned the length penalty between 0.6 and 1.0 (Wu et al., 2016).

5.2 Evaluation

We evaluated our approach for all datasets using ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE-L (RL) (Lin, 2004), which calculate the word-overlap between the reference and the system summaries.

6 Result

Table 1 presents the comparison of R1, R2 and RL scores for Debatepedia dataset. We compared our Recall (R) values of R1, R2 and RL with the works of Nema et al. (2017) and Aryal and Chali (2020). After comparing the results we observed that, our approach successfully achieved new state-of-the-art results for *QFAS* task. We also provided our Precision (P) and F1-measure (F1) values in Table 1.

Model	R1			R2			RL		
	R	P	F1	R	P	F1	R	P	F1
Diversity Model (Nema et al., 2017)	41.26	–	–	18.75	–	–	40.43	–	–
Selective Model (Aryal and Chali, 2020)	43.22	–	–	27.40	–	–	42.73	–	–
Our Approach	**47.16**	**12.38**	**19.23**	**27.48**	**6.71**	**10.58**	**44.07**	**11.48**	**17.91**

Table 1: ROUGE (%) scores of abstractive models on the Debatepedia test set.

Table 2 illustrates F1 values of R1, R2 and RL scores for CNN/DailyMail dataset. After comparing our results with the work of Hasselqvist et al. (2017), we observed that our approach efficiently performed better for CNN/DailyMail dataset.

Model	R1	R2	RL
PG Model (Hasselqvist et al., 2017)	18.25	5.04	16.17
Our Approach	**44.91**	**21.81**	**41.70**

Table 2: ROUGE-F1 (%) scores of abstractive models on the CNN/DailyMail test set.

For Newsroom dataset, no previous *QFAS* work has been performed. Therefore, in Table 3 we present our F1 values of R1, R2 and RL scores for *Abstractive* and *Mixed* datasets of Newsroom for future *QFAS* comparison.

Dataset	R1	R2	RL
Abstractive	15.05	2.26	13.50
Mixed	40.67	22.66	36.92

Table 3: ROUGE-F1 (%) scores of our approach on the Newsroom test set.

7 Conclusion

In this research, one of our aim was to incorporate query and prepare two datasets which contain multiple sentence summaries for *QFAS* task. Our another goal was to pre-process the source documents with a new document sorting approach and then fed to the pretrained model. We targeted to fine-tune the *BERTSUM* model for our *QFAS* task. We compared our results and investigated that our *QFAS* approach successfully achieved new state-of-the-art results for Debatepedia and CNN/DailyMail datasets. As no previous research used Newsroom dataset for *QFAS* task, we provided our results of Newsroom dataset for future comparison of the related research work.

Acknowledgments

We would like to thank the anonymous reviewers for their useful comments. The research reported in this paper was conducted at the University of Lethbridge and supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada discovery grant and the University of Lethbridge.

References

Chudamani Aryal and Yllias Chali. 2020. Selection driven query focused abstractive document summarization. In *Advances in Artificial Intelligence*, pages 118–124, Cham. Springer International Publishing.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Distraction-based neural networks for modeling documents. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 2754–2760, New York, New York, USA. AAAI Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. 2016. Learning-based single-document summarization with compression and anaphoricity constraints. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1998–2008, Berlin, Germany. Association for Computational Linguistics.

Elozino Egonmwan and Yllias Chali. 2019. Transformer-based model for single documents neural summarization. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 70–79, Hong Kong. Association for Computational Linguistics.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Johan Hasselqvist, Niklas Helmertz, and Mikael Kågebäck. 2017. Query-based abstractive summarization using neural networks. *CoRR*, abs/1712.06100.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. Lcsts: A large scale chinese short text summarization dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1967–1972, Lisbon, Portugal. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Preksha Nema, Mitesh M. Khapra, Anirban Laha, and Balaraman Ravindran. 2017. Diversity driven attention model for query-based abstractive summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1063–1072, Vancouver, Canada. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, Las Vegas. IEEE.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1171–1181, Vancouver, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.